Introduction of temporal and spatial dimensions into a recommender system

Master's Thesis submitted

to

Prof. Dr. Ostap Okhrin

Humboldt-Universität zu Berlin
CASE - Center of Applied Statistics and Economics
Ladislaus von Bortkiewicz Chair of Statistics

by

Pierre Navarro
(570525)

in partial fulfillment of the requirements
for the degree of
Master of Science in Statistics
Berlin, October 30, 2015

Abstract

This paper deals with a predictive modelling problem and presents the tools and the approach to tackle it. More precisely, it starts out from a recommender engine operating in supermarkets, and especially from the model predicting whether the customers will use the discount they got from a coupon. This project aims to introduce temporal and spatial dimensions into this model, since customers may have different behaviour according to these two aspects. Improving a predictive model first means to set the proper indicators in order to evaluate its performance. Different metrics are thus introduced on that purpose and selected mainly regarding their suitability towards our business problem. Furthermore, the main highlight of this thesis is made on comparing different models by assessing their predictive power. The temporal and spatial dimensions are successively introduced, by modifying the inputs of the current model and keeping the same method. Through the performance indicators previously defined, we assess whether a new dimension is worth being kept, while getting some insights about the customers' behaviour.

Contents

List of Figures

List of Tables

1 Introduction

Recommender engines have become more and more widely-spread among commerce and e-commerce. In order to win the loyalty of the customers and most importantly to make them buy more products, their purposes are first to recommend the right products to the right customers and then to give them a sufficient incentive to buy them. However, they should not incite the customers too much by giving out too high discounts on the recommended products, at the risk of setting the implied costs too high. That is why recommender engines have to integrate a budgetary limit in their algorithm so that the direct costs implied by the price incentives remain sustainable and manageable. To do so, it is then necessary to get an idea on how many customers will use their discounts and which price-off they have. Ultimately, the goal is to predict as accurately as possible the redemptions.

The starting point of this thesis is a recommender engine which is already operating in some supermarkets. It has been working for some months already so by looking at its data and analysing them, the goal is to better understand the customers' behaviour and to improve the algorithm by increasing its predictive power. Although several methods are worth considering for improving a predictive model, this thesis focuses on taking into account new dimensions within the algorithm. These dimensions, which were so far ignored in the coupon attribution process, are the temporal and the spatial aspects.

The main theoretical challenge of this thesis is then to understand correctly the model and to select appropriate measures to evaluate its performance. That is why this paper clearly presents the business issues and the objectives. Then, it introduces the needed theoretical background related to predictive modelling evaluation in order to complete successfully and properly the given goals. After introducing the data, it presents in details the approach and finally the results: first regarding the time dimension and then concerning the spatial aspect.

2 Context and objectives

2.1 The recommender engines

A recommender system can be defined as an algorithm that attributes to every entity a rating towards each product. It provides a mapping of the customers' preferences. Thus, the algorithm is then able to fulfil its goal: 'recommending' the right products to each entity.

Such engines are more and more common in lots of marketplaces. A typical example is for on-line retailers: when visitors see a product, they can also see a list of related products based on what people having bought the same product also bought. Their goal is clearly to make the customers buy more.

However, the case of on-line marketplaces is somehow simpler than for off-line retailers. Indeed, on-line retailers have at their disposal a lot of data about customers and they exactly know which products each customer bought and saw on their website. For off-line retailers, such a data collection is not as straightforward and even possible to reproduce as a whole since they cannot directly get the data about the purchases of the customers. One alternative to collect some data is to institute a loyalty card.

This is precisely how *our* recommender engine works. It is intended only for the holders of the loyalty card of the retailer. Concretely, these customers have the possibility to go to a specific terminal in some shops before going shopping. By scanning their loyalty card, they get a coupon containing discounts for some specific products available in the supermarket.

The bet is then that the customer will buy the products and get used to them so that afterwards, the products will remain in his/her cart every time he/she comes to the retailer. Then, the recommender engine can be seen as an investment to make customers buy more products: at the first purchase, it represents a cost because of the discount on the products but it becomes beneficial if the customers keep on buying them. The recommender engine is also a way to make customers loyal and to dissuade them from going to competitors by offering them some discounts as soon as they come.

2.2 The introduction of a recommender engine in a chain of supermarket

As explained above, the starting point of this study is a recommender engine for holders of a loyalty supermarket card[1]. This recommender engine was calibrated after several experiments which took place at the beginning of this year during few months. These experiments consisted in giving some discount coupons to the holders for specific products. The discount percentage was randomly attributed (10%, 20%, 30% or 40%). Also the products (called as campaigns) were randomly attributed. Then, it was observed whether the customer uses the coupon or not, which we call the observed redemption.

This recommender engine have been first introduced in 31 different shops in Berlin and its surrounding area. In total, these experiments led to more than 310,000 observations. An observation corresponds to one price-off for a specific product on a coupon.

Finally, all the data about these experiments are stored into a database where the basket of the customers is reported together with the coupon, whether he/she used it or not and lastly some holders' characteristics. The purpose of this database is obviously to learn about

[1]This supermarket chain wants to remain anonymous.

the behaviour of customers in order to improve the recommender engine.

2.3 Calibration of the discount coupon

Once the recommender engine is set up so that it attributes the right product to the right customers, the challenge is then to attribute the right discount to make the customer use it.

There are opposite constraints which interact here:

- The customers, at least some of them, should have a sufficient incentive to use the coupon. In other words, the price-off has to be high enough so that the customers are willing to buy the product on the coupon. Otherwise, the coupons are simply not used, the customers have no chance to buy it again and the machine appears to be useless.
- The discount does not have to be too high. Indeed, the coupon, if it is used, represents a cost for the retailer. The product is not sold at its normal price but at a lower price and often at a loss. So the retailer should not give too many high discount coupons otherwise it will simply loose too much money.

So here is the contradiction the retailer has to face. The recommender engine is an investment to make the customers more loyal and above all to make them buy more. But this has a cost and this cost should not overpass a certain threshold: the budgetary constraint.

That is why the retailer needs to predict the redemption of the coupons and then to calibrate the model so that it respects its budgetary constraint. Predictions are here the key of the problem. If they are not accurate enough, then:

- They can overestimate the redemptions and consequently make the recommender engine underperform in the sense that the discounts would be too low to encourage people to use the coupons. So, the recommender engine would not be used at its full possibilities and the customers would not be incited to use it again and to continue to go to this retailer. The initial goal would not be achieved.
- On the opposite, they can underestimate the redemptions and consequently attribute too high discounts. Finally the budgetary constraint may not be respected.

The challenge is therefore to make the predictions as accurate as possible to avoid any of these two risks.

More formally, the probability of using the coupons is modelled through a logistic regression, according to the price-off and the consumer properties.

If we note y the variable of interest, i.e. the dummy variable for the redemption, whether the coupon was used, o the amount of the discount and π some properties about the customer, then the model is the following:

$$p(y = 1|o, \pi) = \frac{1}{1 + e^{-(\alpha+\beta^1 o+\beta^2 o\pi)}}$$

Then, the price-off discrimination is obtained by maximising $\Sigma_n p(y_n = 1|o_n, \pi_n)$ under the budget constraint $\Sigma_n o_n < B_c$.

2.4 Objectives

The objective of this study is then to improve the model predicting the redemptions in order to both optimize the recommender engine and to satisfy the budgetary constraints.

In order to improve the predictions of the model, there are several options:

- Trying other models and compare the accuracy of the predictions
- Taking into account other parameters that may influence the redemption.

This work focuses on the second option. More specifically, it attempts to introduce two new dimensions that were not in the model so far:

- The time dimension
- The spatial and geographical dimensions.

2.4.1 The time dimension

First, the time matters in consumer choices. It makes sense to think that according to the period of the year, some products are more likely to be bought by customers: chocolates before Easter or Christmas, barbecue meat during summer and pens and papers in September. In the case of these specific examples, people would be more likely to use the coupons that the recommender engine attributes to them. Thus, the time within the year matters.

The day of the week is also a parameter of interest. Indeed, people buy different things according to whether they go shopping on a Monday or on a Friday. It is plausible that at the end of the week for instance, beers, alcoholic beverages and snacks are more susceptible to be bought since people usually go out mostly on week-ends.

Finally, the moment of the day can also be a parameter of interest. Thinking about Saturdays for instance, products of first necessity may be bought more on the morning whereas items for going out and celebrating are more likely to be bought during the evening. A similar statement may apply for the days of the week since during the day, not the same kinds of customers go shopping as during the evenings, right after the office hours.

So all these three time dimensions may allow to estimate more precisely the probability that customers use discount coupons.

Unfortunately, the experiment we are getting the data from did not last for a long time but only for few months. Thus we cannot even consider the first mentioned dimension which is the moment of the year, since we have data only for a couple of months.

Regarding the other dimensions, below is an example from the dataset, which argues in the sense that the time matters when analysing the customers' behaviour. This graph shows the proportion of customers who ever bought woman products among customers coming to the supermarket and scanning their loyalty card.

We clearly see from this graph that on the mornings, this type of customer is over-represented whereas in the evenings, it is always under-represented.

2.4.2 The geographical dimension

Then the geographical dimension can also matter in analysing the customers' behaviour.

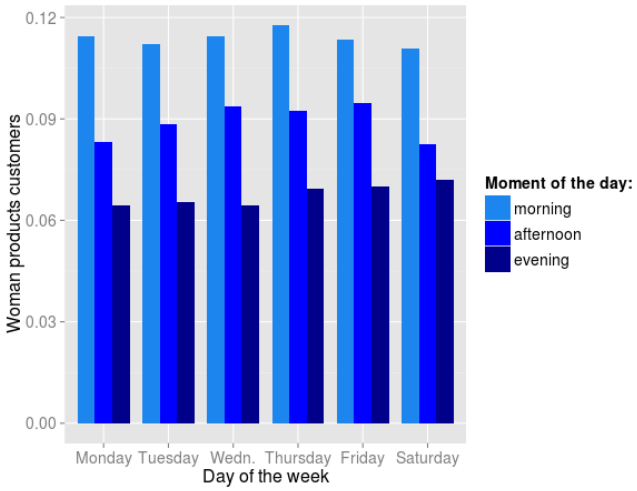

Figure 1: Proportion of woman product customers among loyalty card holders in the supermarkets

First, according to the surroundings, the neighbourhood, the population coming, etc, the purchases and the customer preferences in a specific store may be different from another one. In a neighbourhood where mostly young people live, the consumption and the interest for *Coca-Cola* products must be higher than in a residential area where the proportion of retired people is quite high for instance. The proximity of competitors can also be taken into account. It is likely that in highly-competitive areas, meaning areas with a lot of supermarkets and hypermarkets, the incentive needs to be stronger than in others where the considered store is preserved from any competitor in the near surroundings.

Finally some characteristics of the shop should be considered as they may influence customers' preferences. Indeed, the available sections and services may influence the reason for customer to come to this specific store and so their preferences.

The purpose of this thesis is to see which of the previously mentioned parameters do influence the probability of redemption for the loyalty card holders.

3 Theoretical background

Now that we have incorporated this work within the framework of the business logic, statistical tools and methods that may be useful to put it into practice will be introduced.

3.1 The logistic model

As described previously, the business problem is to predict a customer behaviour that can be summarized in a binary variable: whether the customer used the coupon or not.

With a binary outcome, the first model that comes in mind is obviously the logistic regression. This is actually the model used in the recommender engine. As a recall, the logistic function is defined as follows:

$$F(t) = \frac{1}{1 + e^{-t}}$$

The logistic regression equation is thus:

$$p(y = 1|x) = F(\beta_0 + \Sigma_n \beta_n x_n) = \frac{1}{1 + e^{-\beta_0 - \Sigma_n \beta_n x_n}}$$

The parameters are then estimated, usually through maximizing the likelihood.

Some of the main assets of the logistic regression which justify why this model was chosen are:

- its scalability: this model is rather easy to compute and do not need a lot of resources, even with a big number of observations
- its comprehensibility: the logistic model is very widely-spread, very understandable and the output of such a model is quite easy to interpret.

About this last point, due to the form of the inherent equation, the coefficients of a logistic regression are not directly interpretable - in comparison with linear regressions for instance. Indeed, only their sign allows to say whether they have a positive or negative influence on the dependent variable. That is why odd-ratios are pretty useful to get a quantitative interpretation for the coefficients.

The odds are a relative probability to reflect the likelihood that an event will happen. In the case of a binary variable, we divide the probability of success by the probability of failure. An odd-ratio is simply a ratio of two odds, comparing the odds of the same event in two distinct environments. The odd-ratio is defined as follow:

$$OR(z_1, z_2) = \frac{ODDS(z_1)}{ODDS(z_2)} = \frac{\frac{F(z_1)}{1-F(z_1)}}{\frac{F(z_2)}{1-F(z_2)}}$$

Taking now z and $z + 1$ to see the effect of increasing the independent variable by one unit on the likelihood of the dependent variable:

$$OR(z + 1, z) = \frac{ODDS(z + 1)}{ODDS(z)} = \frac{\frac{F(z+1)}{1-F(z+1)}}{\frac{F(z)}{1-F(z)}}$$

By replacing $F(.)$, we get:

$$OR(z) = \frac{e^{\beta_0 + \beta_1 z}}{e^{\beta_0 + \beta_1 (z+1)}} = e^{\beta_1}$$

This result can be easily generalised for a logistic regression with n independent variables with $OR(z_i) = e^{\beta_i}$.

Thus, we can deduce from the coefficients we get from the logistic regression a quantitative interpretation. This will be actually useful to understand what influence the customers regarding the redemption and how. However, this will not be helpful to detect any improvement of the model in terms of predictive accuracy.

3.2 The evaluation of the model predictive power

Now that we have set up the goal of this work as improving a predictive model, it is necessary to look into how to measure whether the model is improved or not. *How to evaluate the model?* To be more precise: *how to assess the predictive power of a logistic regression?*

3.2.1 How to test the predictions?

First, for such an assessment and to get an idea of the performance of the model, we need to test it on observations for which we already know the output. In other words, we need to run the estimated model in order to get the predictions for some data points and compare them with the real observations.

The way to go is usually to split the whole sample into two subsets: one to estimate the model and the other one to test it. Using the same data for training the model and testing it would not be so rigorous and we actually have at our disposal quite a big sample which allows large training and testing sets.

Regarding how to split the model and which share of observations to put aside for testing, there is no strict rule about that. However, it is better to keep most of the observations for the model estimation (training). Usually, 70 to 80% of the observations are dedicated to the model estimation whereas the remaining 20-30% are integrated into the test set. This depends then on the number of observations. Too many observations may obliged the user to reduce the share of training data to avoid too high computational costs and on the contrary too few observations may force to increase it to get a consistent estimation.

Another approach to test the model would be to go for a N-fold cross classification. The principle is to split the database into N sub-samples, to train the model on $N-1$ subsets and to evaluate it on the remaining one, and finally to repeat this process N by changing the validation set. This method is actually quite costly in terms of resources but it allows more robust estimations.

3.2.2 The accuracy paradox

Once we have the two sets, we estimate the model on the training data and estimate the predictions on the test data. *How then to measure the efficiency of the model?*

Usually, three different types of metrics can be distinguished in terms of predictive model evaluation:

- The threshold metrics, which assess the predictive power of not only a model, but also a threshold. They assess how close the predicted classes are to actual classes, how accurate the predicted class are given a specific threshold.
- The ranking metrics, which assess the predictive power of a model, not taking into account any threshold. More precisely, they look at the rank of the predicted probabilities and see the coherence with the real classes of the test observations.
- The probability metrics also assess the model without taking into account any threshold. They compare the output probabilities with the actual class of the test observations.

The most basic and intuitive approach to measure the efficiency of a predictive model would be to consider the accuracy of the model, more specifically called the percentage of correctly classified (PCC). It is defined as the number of accurate predictions over the total number of predictions. This is a threshold metric since it takes the predicted classes and not the predicted probabilities, already assuming a certain threshold.

$$Acc = PCC = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{y_i=\hat{y}_i}$$

With y the actual observation, $\hat{y}$ the predicted class and n the number of observations.

We now introduce the confusion matrix and rank the predictions accordingly:

		Actual class	
		0 (negative)	**1 (positive)**
Predicted class	**0 (negative)**	True negative (TN)	False negative (FN)
	1 (positive)	False positive (FP)	True positive (TP)

Table 1: The confusion matrix - General form

According to this confusion matrix, we can rewrite the accuracy as:

$$PCC = \frac{TP + TN}{TP + TN + FP + FN}$$

The accuracy has the advantage to be easily understandable and simple to compute. However, the model predicts probabilities and not directly the classes. Then, *how to transform these probabilities into a binary output? By simply rounding the probabilities?* There is a need for defining a *decision threshold*, also called a *cutoff*.

We consider a case with very unbalanced classes. This means that a huge share of the observations in the training sample belongs to one class. We consider that this class is the negative one. Then, the probability of success will be pretty low. Rounding the probabilities will probably lead to get only negative predictions, as shown in the confusion matrix below. Rounding actually means taking a decision threshold of 0.5.

		Actual class	
		0 (negative)	**1 (positive)**
Predicted class	**0 (negative)**	1,950	50
	1 (positive)	0	0

Table 2: The confusion matrix - Accuracy paradox

The accuracy for such prediction is equal to 97.5% which seems very promising. Considering another decision threshold on the same predictions, lower than the previous one, we obtain the following confusion matrix:

The accuracy for the second threshold (92%), is lower than for the first one. However, the first threshold would be completely useless as it predicts only negative cases. So there is no point in using it. This is actually the *accuracy paradox*.

Globally, the accuracy is not a very good and reliable measure for predictions, especially where the underlying variable is very unbalanced. Indeed, if this variable is class unbalanced,

		Actual class	
		0 (negative)	**1 (positive)**
Predicted class	**0 (negative)**	1,800	10
	1 (positive)	150	40

Table 3: The confusion matrix - Accuracy paradox

then the risk is that all the predictions will take the value of the largely-represented class so that the accuracy will be super high.

3.2.3 More indicators to evaluate the predictive power

Thus, there is a need to introduce further concepts since the predictive power of a model cannot be summarize by the accuracy indicator.

Two of them, widely-used, are the specificity and the sensitivity. These indicators are also based on the confusion matrix:

- The specificity, also called the true negative rate (TNR):

$$TNR = \frac{TN}{TN + FP}$$

- The sensitivity, also called the true positive rate (TPR):

$$TPR = \frac{TP}{TP + FN}$$

A good predictive model obviously gets a high specificity as well as a high sensitivity.

For the first above confusion matrix example, the specificity would be 100% but the sensitivity would be 0%. Therefore, we can see that the accuracy does not reflect the complete predictive power of the model.

For some specific works, the goal can be to specifically maximize one of the two indicators. If we consider the example of a clinical test where positive would mean that the patient has the disease, the main goal is clearly to minimize the number of false negative. Therefore, the decision threshold would be the one maximizing the sensitivity.

However, for this project, we do not have such a precise goal. The goal is then somehow to maximize both of the indicators since no misclassification has a higher cost than the other.

Similarly to specificity and sensitivity, we define the false positive rate (FPR) and the false negative rate (FNR):

- The false negative rate:

$$FNR = \frac{FN}{FN + TP} = 1 - sensitivity$$

- The false positive rate:

$$FPR = \frac{FP}{FP + TN} = 1 - specificity$$

These rates are obviously to minimize. The lower they are, the better the model is.

As for the specificity and the sensitivity, there is no precise target regarding our model. Again, a clinical study must have to focus on the FNR to avoid too many cases of undetected carrier of a certain disease. However, for the redemption predictions, we want to minimize both of them, without a specific distinction.

3.2.4 The ROC curve

Now that the different indicators were introduced, the Receiver Operating Characteristic (ROC) curve can be mentioned.

This curve is actually the representation of the predictive performance of a model, taking into account an infinity of classifiers, meaning changing the decision thresholds separating positive predictions from negative ones.

This curve illustrates the relationship between sensitivity and specificity. Indeed, on the ROC curve:

- the x-axis represents the FPR (1 - $specificity$)
- the y-axis the TPR ($sensitivity$).

Then, each point represents a different classifier of the model, obtained by changing the decision threshold and so the prediction distribution.

On such a graph:

- The $(0, 0)$ represents a decision threshold equal to 1. Then, all the predictions are negative.
- The $(1, 1)$ represents a decision threshold equal to 0. Then, all the predictions are positive.
- The $(0, 1)$ is the ideal point. All the positive predictions are true positive and none of them is wrongly positive and thus the same goes for negative ones.
- The diagonal represents the random guess, with randomly distributed predictions. In this case, the rate of true positive is equal to the rate of false negative.

A curve going below the diagonal would mean that the predictions are worse than the random guess and so perverse. It would say that predictions are opposite to the true class since the rate of false positive is higher than the rate of true positive.

On the opposite, a curve above the random guess means that the predictions are better since the true positive rate is above the false positive one. And as long as the curve gets closer to the $(0, 1)$ point, the model keeps improving.

Below is an example of a ROC curve from a model which performs better than chance.

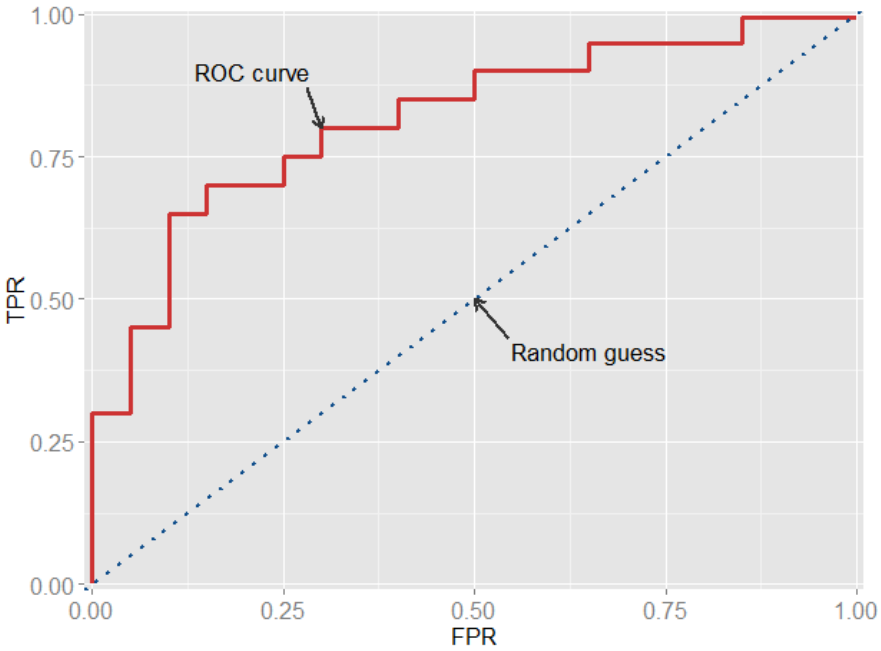

Figure 2: Construction of the ROC curve

The ROC curve can have different purposes:

- Determining an optimal decision threshold, minimizing the FPR and so maximizing the TPR
- See whether a specific classifier is better than pure chance
- Comparing different classifiers.

Regarding the purpose of this project, the last option is the one that will matter more here.

The below graph illustrates this purpose with two hypothetical models that we call red and blue, according to the color of their ROC curves. On the red area, the red model performs better than the blue one and vice-versa. So, for some decision thresholds, the blue model performs better and for some other cutoff points, the red one gives better predicted classes.

The visual comparison is of course possible. What intuitively matters with this ROC curve is its proximity to the up-left corner and a first ROC curve above a second ROC curve would simply mean that the first model is better than the second one.

However, the visual comparison is not always very comfortable. For instance, going back to the example of the red and blue models, it is hard to say at a simple glance on the graph which model is overall the best one for predicting.

So, there is a need to find a good indicator to avoid this visual comparison which might be delicate and not so straightforward and rigorous. Some indicators and measures exist to assess the predictive performance of a model through its ROC curve.

First, we could take the point that minimize the distance with this optimal point:

$$min(\sqrt{(-fpr)^2 + (1 - tpr)^2}) = min(\sqrt{fpr^2 + fnr^2})$$

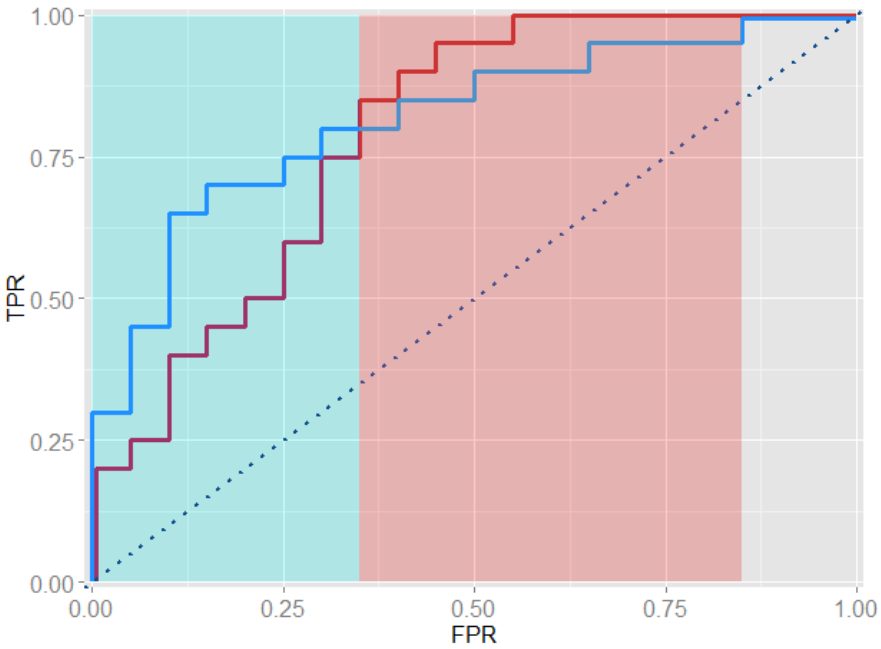

Figure 3: Example of classifier comparison through the ROC curve

Nonetheless, this approach is not widely-used. Instead, in practice, the Area Under the Curve (AUC) is computed. Intuitively, it is easily understandable that the bigger is this area, the more predictive power the model has. Indeed, the closer the curve is from the $(0, 1)$ point, the bigger the area under it will be. And between two models, the one which has its ROC curve lying above the other automatically has its AUC also higher.

This AUC is always between 0 and 1 since the curve is located within the unit square. An AUC equal to 0.5 can have several meanings:

- The predictive model attributes the same prediction to every test observation, positive or negative. This would mean that the ROC is the diagonal of the unit square.
- The predictive model attributes predictions so that the two different classes have the same distribution. This results in a ROC close to the diagonal.
- The predictive model gives 1 to half of the two classes, and 0 to the other half.

Therefore, an AUC of 0.5 does not necessarily mean random predictions. It should be higher than 0.5 to say that the model is better than the random guess. The closer it is to 1, the better it is.

Finally, the AUC measures the probability that a random positive (from the class 1) observation is ranked higher than a random negative (from the class 0) observation. The AUC is a ranking metric; it takes into consideration all the possible configurations in terms of decision threshold.

3.3 The Mann-Whitney U test

It is sometimes said that the AUC is the normalised version of the Mann-Whitney U test. This part aims to explain why in order to better understand the concept of the AUC.

The principle of the test

The Mann-Whitney U test is a non-parametric test comparing the distribution of two samples. The null hypothesis of this test is that two sub-samples have the same distribution and then come from the same population.

Supposing that N is the number of observations in the whole sample (combining the two sub-samples), n_1 the size of the first sub-sample and n_2 the size of the second one, the test works as follows:

- It first ranks the N observations from both samples, attributing 1 to the lowest value and N to the largest one.
- It then computes the U statistics for both sub-samples: $U_i = R_i - \frac{n_i(n_i+1)}{2}$. R_i is the sum of ranks in sample i.
- Summing up R_1 and R_2 gives:

$$R_1 + R_2 - \frac{n_1(n_1+1) + n_2(n_2+1)}{2}$$

Since $R_1 + R_2$ is by definition the sum of the ranks of the all sample, it is equal to $\frac{N(N+1)}{2}$. We have then:

$$U_1 + U_2 = \frac{N(N+1)}{2} - \frac{n_1(n_1+1) + n_2(n_2+1)}{2}$$

And $U_1 + U_2 = \frac{N(N+1)}{2} - \frac{n_1^2+n_2^2+N}{2}$ since $n_1 + n_2 = N$. Yet:

$$\begin{aligned} U_1 + U_2 &= \frac{N(N+1)}{2} - \frac{(n_1+n_2)^2 - 2n_1n_2 + N}{2} \\ &= \frac{N^2+N}{2} - \frac{N^2+N}{2} + n_1n_2 \\ U_1 + U_2 &= n_1n_2 \end{aligned}$$

Under the null hypothesis, we have:

$$U_1 = U_2 = \frac{U}{2} = \frac{n_1n_2}{2}$$

For large samples, U_1 is normally distributed, with $\frac{n_1n_2}{2}$ as mean.

The link with the AUC

We now consider a predictive model for a binary variable. The testing of this predictive model on a sample of observations gives a set of probabilities which can be interpreted as probabilities of success, along with the binary variable indicating whether there is actually success or not.

We rank the probabilities in a descending order. Suppose that we have N tested observations in total, n_1 are success and n_2 are not a success. Here is a made-up example, with $N = 17$, $n_1 = 8$ and $n_2 = 9$. The *success* class is the 1-class:

Observation	**Real class**	**Probability of success**
12	1	0.975
3	1	0.924
5	0	0.843
9	1	0.754
2	0	0.723
15	1	0.687
17	1	0.674
11	0	0.621
8	1	0.543
10	0	0.443
1	0	0.432
6	1	0.342
7	1	0.301
14	0	0.219
4	0	0.143
16	0	0.126
13	0	0.045

Table 4: Example - Sorted data

Now, consider we start with a decision threshold of 1, then we have a FPR and a TPR equal to 0 since there is no predicted success. This point is at the bottom-left of the ROC area, $(0, 0)$.

We now decrease the threshold so that we include the first success observation in the positive group (meaning the success with the highest predicted probability) without including no-success observation. Then, the TPR is now equal to $\frac{1}{n_1}$ and the FPR still equal to 0. The gained area is then equal to $\frac{1}{n_1}$.

If by increasing the decision threshold and thus including one more success observation, we also include x no-success observations. The TPR increases by $\frac{1}{n_1}$ and the FPR increases by $\frac{x}{n_2}$. Then the gained area is $(1 - \frac{x}{n_2}) \times \frac{1}{n_1}$.

So, each time we decrease the thresholds to include one more success observations into the success group, we get:

$$Gain = \frac{n_2 - x}{n_1 \times n_2}$$

x being the number of no-success observations being accidentally moved into the predicted success class. Below, the graph allows to visualize the gain (red rectangle area). The height of the rectangle is $\frac{1}{n_1} = \frac{1}{8}$ and the length $\frac{x-n_2}{n_2} = \frac{8}{9}$:

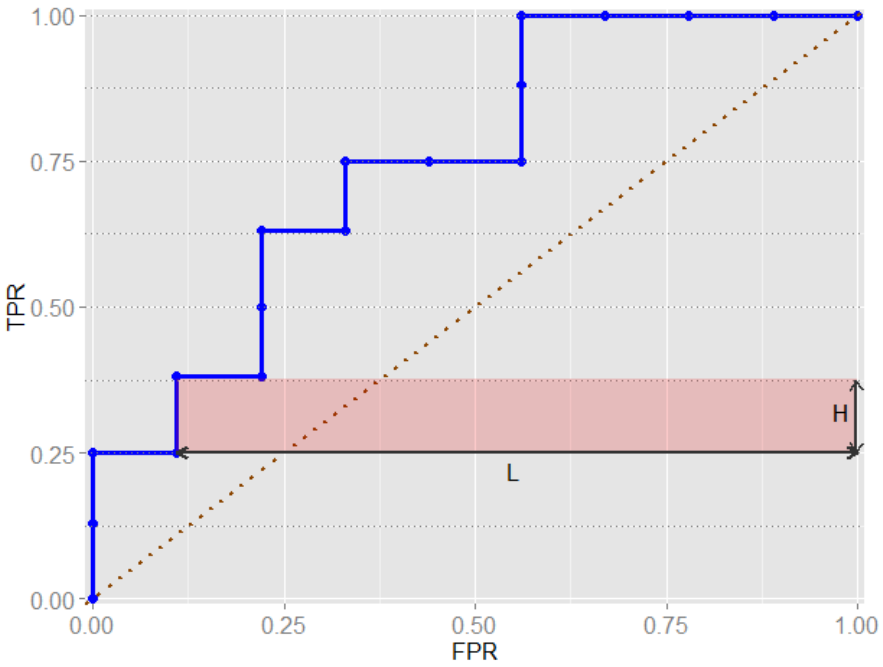

Figure 4: Construction of the ROC curve

Then, the total area under the AUC is calculated as follows:

$$\begin{aligned} AUC &= \frac{1}{n_1}\sum_{i=1}^{n_1}\frac{(n_2 - x_i)}{n_2} \\ &= 1 - \frac{1}{n_1 n_2}\sum_{i=1}^{n_1} x_i \end{aligned}$$

Furthermore, if r_i is the rank of the success observation being included in the predicted success class, we get:

$$x_i = r_i - i$$

Thus:

$$\begin{aligned} AUC &= 1 - \frac{1}{n_1 n_2}\sum_{i=1}^{n_1}(r_i - i) \\ &= 1 - \frac{1}{n_1 n_2}(\sum_{i=1}^{n_1} r_i - \frac{n_1(n_1+1)}{2}) \end{aligned}$$

Reminding from the Mann-Whitney U test that $U_1 = \sum_{i=1}^{n_1} r_i - \frac{n_1(n_1+1)}{2}$, we get:

$$\begin{aligned} AUC &= 1 - \frac{1}{n_1 n_2} U_1 \\ U_1 &= n_1 n_2 (1 - AUC) \end{aligned}$$

3.4 The Gini coefficient

Finally, we introduce a last ranking metrics which is also linked with the AUC.

The Gini coefficient was first introduced in economics, as an indicator of income inequality. As the AUC, the Gini coefficient is related to the area under some curves.

The first curve is the Lorenz curve, known as the representation of the cumulative distribution function of income. The second one in the diagonal ($y = 0.5x$) which represents actually an income distribution in a world where all the people would earn the same income. This line is also called the 'line of equality'.

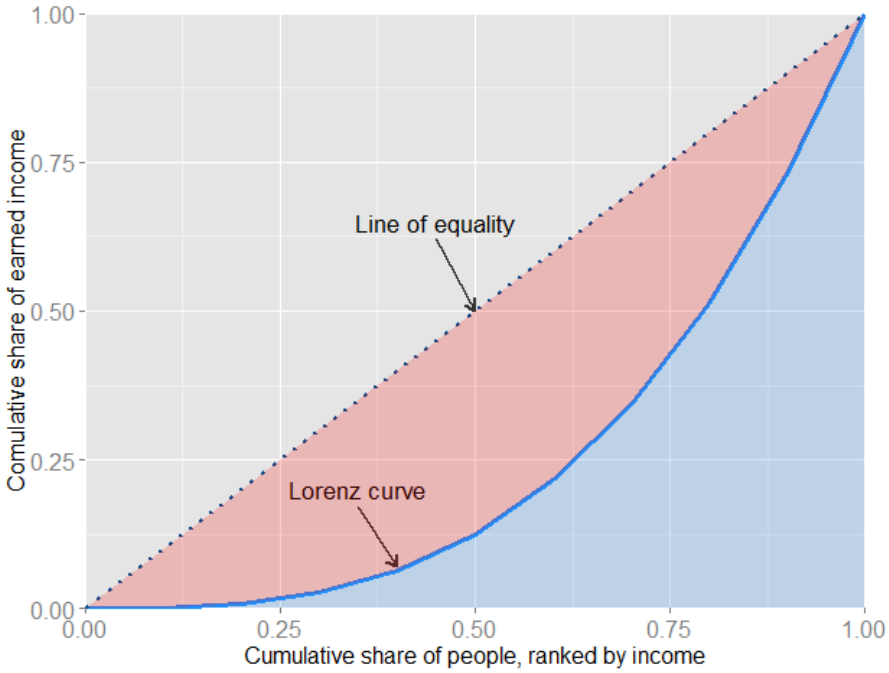

Figure 5: Lorenz curve and line of equality

Considering A^L, the area under the Lorenz curve (in blue on the graph) and A^E the area under the line of equality (cumulated blue and red areas), the Gini coefficient is:

$$G = \frac{A^E - A^L}{A^L} = \frac{0.5 - A^L}{0.5}$$

The higher the value of the Gini coefficient is, the more unequally distributed the incomes are.

Instead of ranking the people from the poorest to the richest, we rank them from the richest to the poorest. Then, the modified Lorenz curve would lie above the line of equality. And the modified form[2] of the Gini coefficient is:

$$G' = \frac{A^L - 0.5}{0.5} = 2A^L - 1$$

[2]The new form is actually the negative version of the initial one, so that the coefficient remains in the interval $[0, 1]$.

Therefore, the relationship with the AUC is more obvious. The AUC does not represent the cumulative distribution of incomes anymore but the cumulative distribution of true positive predictions among positive observations according to the rate of false positive among negative observations. The line of equality is now the *line of chance*, representing the similarity of the distribution in all the classes.

So quite naturally we get:

$$Gini = 2AUC - 1$$

As a result, using the Gini or the AUC for evaluating a predictive model is equivalent.

3.5 The Brier score

Finally, in order to diversify the evaluation approach to get a better idea of the models' performance, a probability metric is introduced.

One of the most widely-used probability metric is the Brier score. This measure is actually a mean-square error for probability predictions.

$$BrierScore = \frac{1}{n}\sum_{i=1}^{n}(y_i - p(y_i|x_i))^2$$

n is the number of test observations, y_i is the actual class (either 0 or 1) and $p(y_i|x_i)$ is the predicted probability.

The lower the Brier Score is, the better the probability predictions are calibrated so the better the predictive model performs.

Summary

Summarizing the main points of the theoretical background for the next steps of the project:

- To test the models, the split-sample approach will be implemented. Indeed, the N-fold cross-validation requires much more resources, which can become very costly when estimating and evaluating a lot of different models. Furthermore, N-fold approach is not necessarily justified here since the dataset is large enough to get consistent estimated parameters for the model with a simple split-sample approach, avoiding any *lucky-sample* effect. The N-fold approach is actually more used for medium-size databases. The split-sample we use is 70%-30% and it is randomly made.

- As previously mentioned, choosing a decision threshold can be somehow tricky and result in misleading measures, as seen with the accuracy paradox. Moreover, the point of this study is to improve the predicted probabilities, not the predicted classes. Therefore, there is no need to tackle the issue of the cutoff point determination. That is why we will not use any threshold metric to assess the models.

- Instead, a ranking metric will be implemented for every estimated model. The goal is indeed to improve the overall predictive power of the different models. We could

equivalently have chosen the AUC and the Gini, since they are linearly related. We go for the Gini coefficient, mostly for coherence reasons with the currently-used measures for the recommender engine.

- A probability metric, the Brier Score, will also be used in order to get another view on every model performance.

4 Data

4.1 The initial databases

The initial extraction from the database contains 37 variables and more than 314,000 observations. An observation corresponds to a product discount on a printed coupon.

These data directly come from the retailer. This is the database for holders of the loyalty card who used the recommender engine in the first 31 shops where the recommender engine was introduced for the test phase.

In this dataset:

- Several variables allow to identify some consumer properties and to get an idea of his/her purchase frequency: total value/unit sales, number of coupon prints, number of card scans at the till...
- Some variables deal with store information (id, address) and some others are related to the time when the coupon was printed (date, time, day of the the week).
- A couple of variables concern the coupon itself, especially the price-off (in percentage) and the redemption, i.e. whether the customer used the coupon. These last two variables will be the most important ones for the next steps of the work.

A complete list of the variables from this initial database can be found in the appendix.

Updated database for the spatial dimension analysis

For the section about spatial dimension, the data sample is not the same as for the time dimension analysis. Indeed, during the summer, the recommender engine was introduced into 111 new shops in Berlin and its area. Then, this new database gathers data from 142 shops. It was the opportunity to get more data and to have more locations taken into account. That is why we have in this section more observations: 868911.

An updated list of variables is also in the appendix.

Summary

	Table for the time dimension	Table for the spatial dimension
Number of observations	313,850	868,911
Number of variables	37	11
Number of distinct shops	31	142

Table 5: Summary about datasets

4.2 Data treatments

From this subsection on, all the implementations have been done using *R*.

For the temporal analysis

The database contains three time-related variables:

- *printDate*: date of coupon print
- *printTime*: time of coupon print
- *printWeekDay*: day of the week of coupon print

From these three variables, three indicators were built:

- *end_of_week*: dummy variable for coupon print on Fridays and Saturdays
- *saturday*: dummy variable for coupon which were printed on Saturdays.
- *momentDay*: factor variable with the following categories: *morning / afternoon / evening.*
  - *morning* for coupons printed before 12.00
  - *afternoon* for coupons printed between 12.00 and 18.00
  - *evening* for coupons printed after 18.00.

For the spatial analysis

In the database, the only geographical data we have are the name of the shops. This name actually corresponds to the name of the street where the store is located. The first step was then to get the exact address of the shop, with the street number and the zip code.

However, we obviously have no information about nearby competitors. And if we now know the addresses, we do not know much about the areas.

To solve these issues, the *Google Maps API* was a great help to retrieve in an automatic way information about the environment around every shop where the recommender engine is operating: both information about the competitors and details about the area. First, the *Google Maps Geocoding API* was used to transform the addresses into their geographic coordinates (latitude and longitude). From these geographic coordinates, we are then able to use the *Places library* from the *Google Maps API* and more specifically the *nearbysearch*. This tool allows the user to look for specific places in a given geographical area. In the case of this project, all the places classified as *'grocery or supermarket'* were retrieved. Then, the output gives the 20 closest competitors, ranked by distance. Only the ones located in a 1500 meter radius around every shop were kept. For each of the nearby place, we have the corresponding name, the address and the geographical coordinates.

These last parameters allow to then compute the distance as the crow flies between the supermarket with recommender engine and each of the competitors. This distance calculation actually corresponds to the shortest distance between two points on an ellipsoid[3]. And finally, a text analysis on the shop names gives us the name of the chain to know who are the competitors around.

Furthermore, a classification of the competitors was built, according to the following definitions:

[3]The distance as the crow flies was computed through a package for practical application of geodesics on an ellipsoid, taking WGS84 as the reference ellipsoid for the Earth.

- Hard-discount: small or medium-size shop, characterized by very low prices and where only few brands are in stock.
- Supermarket: medium-size shop where all the common products and brands are in stock.
- Hypermarket: large shops with a lot of choices in terms of products and brands, more usually in peripheral areas.

Knowing the chain of every competitor around, it was then easy to allocate one type to every competitor. It is now meaningful to precise that *our* chain belong to the second category.

Now that we have identified and located the competitors around the stores we are interested in, and moreover calculated the distances, we can compute a series of variables/indicators:

- *nb_competitors100*: number of competitors in a radius of 100 meters (also for 200/500/1000/1500 meters)
- *nb_hard_discount100*: number of competing hard-discount stores in a radius of 100 meters (also for 200/500/1000/1500 meters)
- *nb_supermarkt100*: number of competing supermarkets in a radius of 100 meters (also for 200/500/1000/1500 meters)
- *nb_hypermarkt100*: number of competing hypermarkets in a radius of 100 meters (also for 200/500/1000/1500 meters)
- *min_competitor*: distance with the closest competitor
- *min_hard_discount*: distance with the closest hard-discount competitor
- *min_supermarkt*: distance with the closest competing supermarket
- *min_hypermarkt*: distance with the closest competing hypermarket
- *closest_competitor*: type of the closest competitor

Regarding the characteristics of the area of the stores, it was hard to compute a program which gives us a lot of information about it. Still using *Google Maps API*, it was nevertheless possible to retrieve the name of the district (*bezirk*) and then compute a couple of indicators:

- *east_west*: whether the shop was in a former eastern/western neighbourhood.
- *into_ring*: dummy variable, whether the shop is located within the *Ring*, i.e. in a central district of Berlin.

5 The time dimension

As the background of the study is now clear and the data prepared, it is time to start with the core of the topic, first with the time dimension. An exploratory analysis is required in order to point out the main parameters which influence the redemption rate and to somehow guide the introduction of the time dimension into the model.

5.1 Exploratory analysis

First, we clearly see that the redemption rate varies according to the day of the week. On Tuesdays, the redemptions are much rarer than on the ends of the week: 2.7% vs. 3.4% on Saturdays.

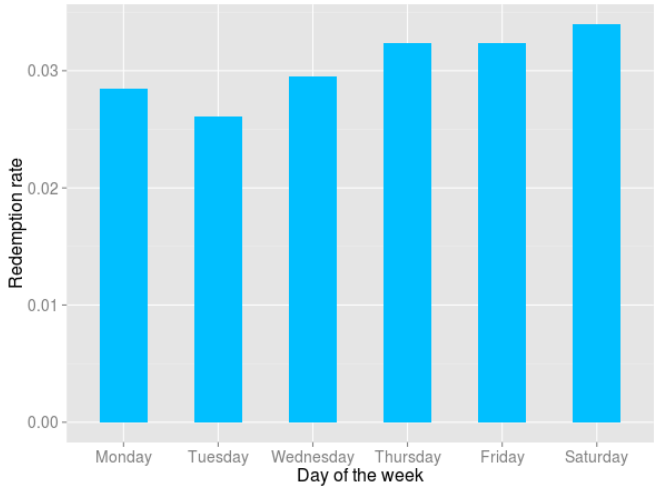

Figure 6: Redemption rate by day of the week

It might simply be that during the end of the week, customers are less in a hurry and spend more time in the shop. Also they are more likely to pay real attention to the coupon, to go to the section where the discounted product is and to consider buying it. Another explanation could be that during the week, more people simply go to the supermarket to buy few things that they miss at home. On the opposite, on the end of the week they go in order to literally fill their stocks and so they buy a larger range of products. Therefore, they are more likely to buy the discounted products.

The redemption rate also changes according to the moment of the day. Below is a graph with the redemption rate broken down by the opening hours of the day. We see that the redemption rate goes up as long as the day goes on. This could be explained with similar reasons as above. On the evenings, people have more time than during the day.

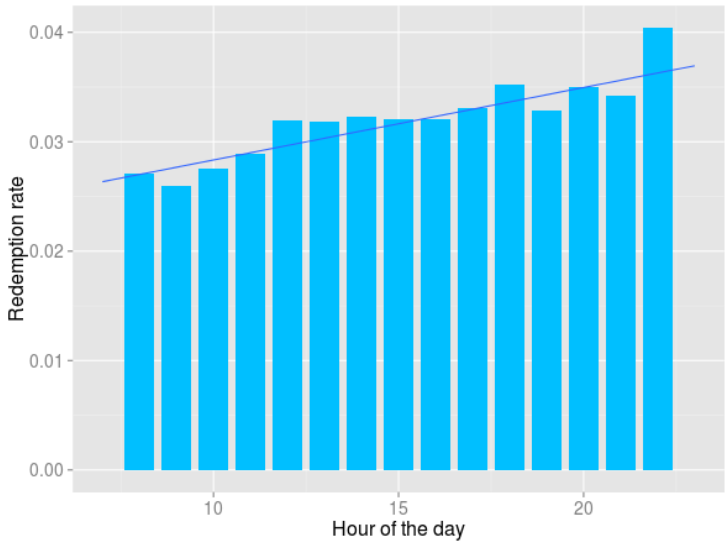

Figure 7: Redemption rate by hour

We now consider the three periods defined as previously: morning, afternoon and evening. On the evenings the redemption rate is 3.50%, whereas only 2.70% of the coupons lead to a redemption on the mornings.

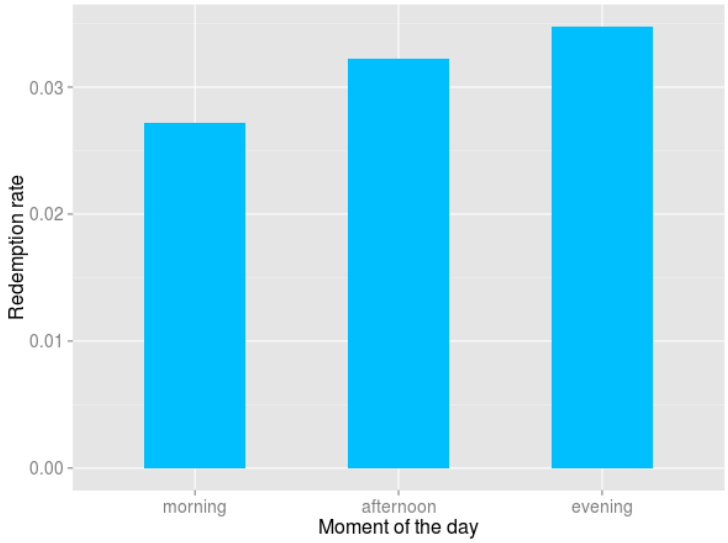

Figure 8: Redemption rate according to the moment of the day

And finally, if we mix the two above graphs, we clearly see that the trend we observed about the moments of the day is obvious for all the days of the week.

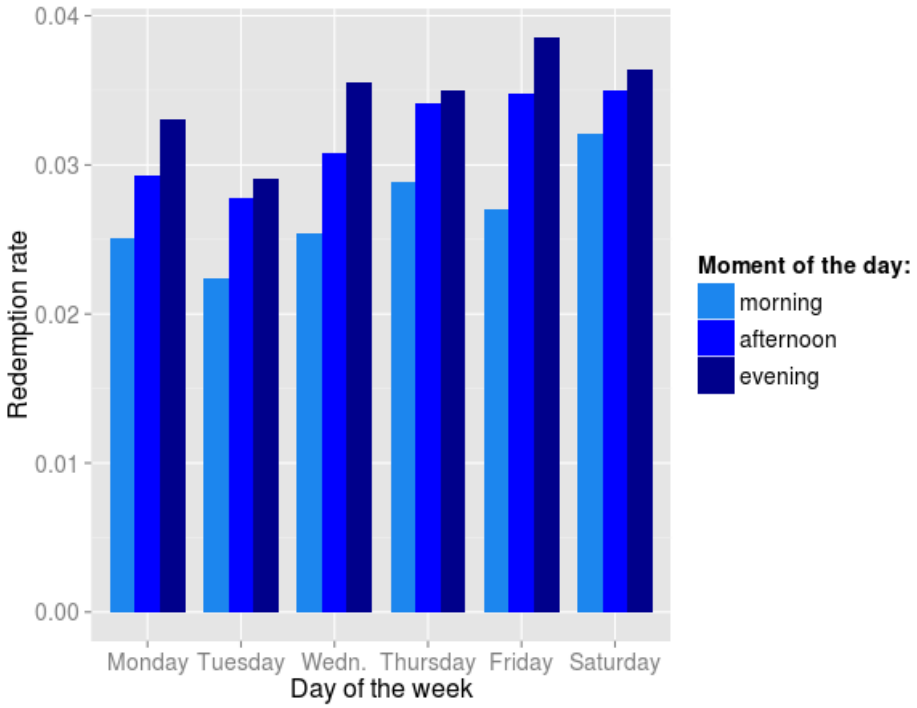

Figure 9: Redemption rate according to the moment of the day and the day of the week

To quickly summarize, the redemption rate is indeed not constant all through the time. It especially varies according to the day of the week and to the moment of the day. It is then conceivable that these two parameters may improve the model.

5.2 Introducing the time dimension into the model

The goal is then to see more formally whether these time-related variables/indicators influence the coupon redemption and whether they can be introduced in the base model to improve it and therefore get better probability estimations. For this purpose, they were included into the base model in order to see whether the evaluation indicators go up.

This method implies a lot of attempts. From the above exploratory analyses, I first tried with the variables which seemed to directly influence the redemption rate. Then, I also tried combining dimensions and variables.

The base model we consider is the following:

Model	**Gini coeff**	**Brier**
$redemption \sim priceOff + cpTop$	0.393	0.030

Table 6: Base model - Time dimension

Its ROC curve is represented on the following graph:

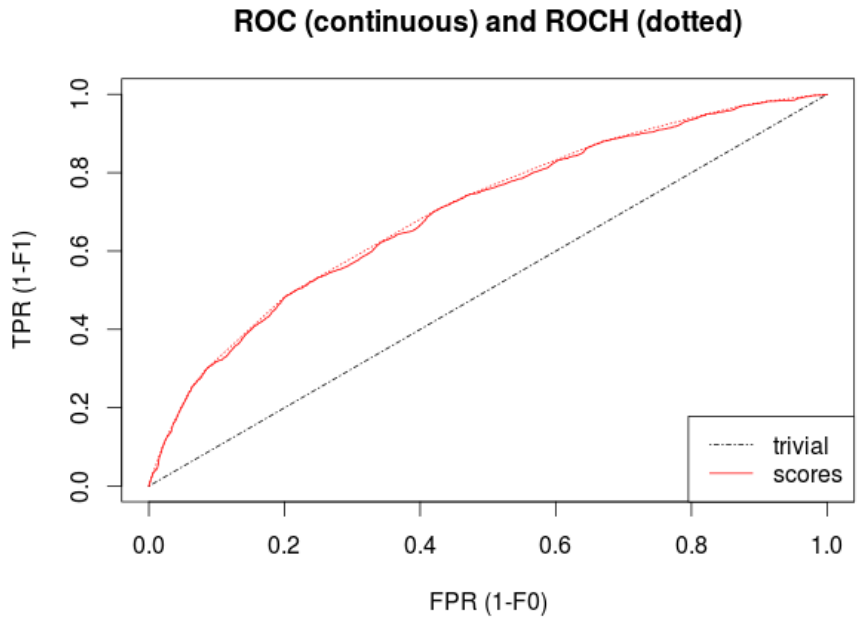

Figure 10: ROC curve for the base model - Time dimension

The introduction of some variables or combinations of variables does not bring enough information to the model to be kept. This is the case for the days of the week. On the contrary, adding together *momentDay* and *saturday*, we see a slight improvement of the Gini coefficient and therefore, of the predictability power of the model[4].

Model	**Gini coeff**	**Brier**
$redemption \sim priceOff + cpTop + WeekDay$	0.393	0.030
$redemption \sim priceOff + cpTop + momentDay$	0.393	0.030
$redemption \sim priceOff + cpTop + saturday$	0.393	0.030
$redemption \sim priceOff + cpTop + saturday + momentDay$	0.395	0.030
$redemption \sim priceOff + cpTop + end\_of\_week$	0.393	0.030
$redemption \sim cpTop + priceOff \times saturday$	0.394	0.030
$redemption \sim priceOff + cpTop + momentDay + WeekDay$	0.396	0.031

Table 7: Unsuccessful models - Time dimension

5.3 Results

After several attempts of this kind, keeping trying to always improve the predictability power of the model, the best one appears to be the following:

The Gini coefficient is indeed slightly higher than for the previous models. However, the Brier score remains quite constant.

[4]For the meaning of the variables, please refer to the Appendix

Model		Gini coeff	Brier
$redemption \sim$	$priceOff \times saturday + cpTop + momentDay + ist + avgUS$	0.410	0.030

Table 8: Best model - Time dimension

The respective ROC curve is represented on the following graph. We cannot visually see any improvement compared to the base model's one, except that the curve is smoother.

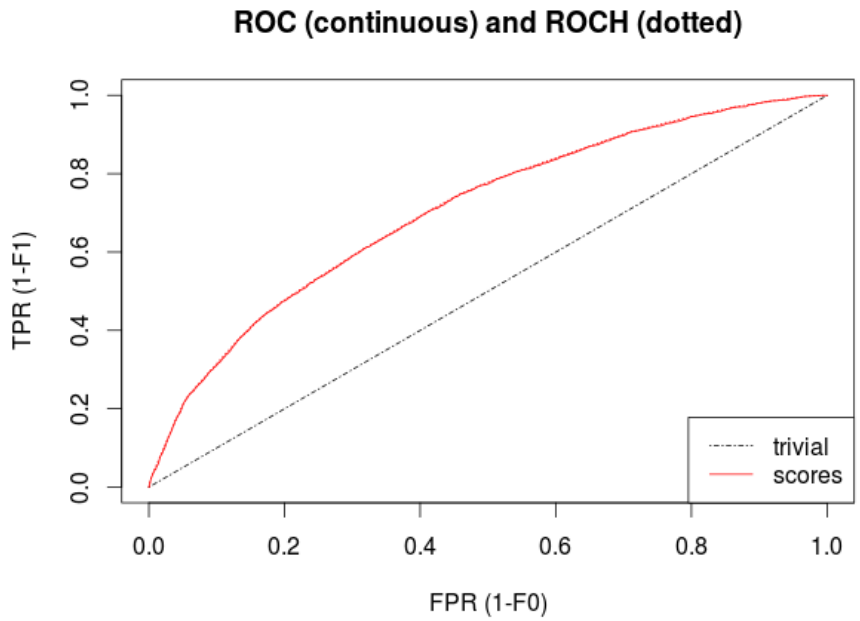

Figure 11: ROC curve for the best model - Time dimension

To better understand the role of each variable, it is interesting to have a closer look at the model itself. The output table for this model is the following[5]:

All the coefficients are significant.

- The coefficient for *saturday* is positive, meaning that redemption is more likely to happen on Saturdays, which is coherent with the above exploratory analysis.
- A similar remark also goes for *evening*, when redemption is more likely than during the afternoons. For the mornings, it is obviously the opposite with a negative coefficient, also as pointed out above.
- The positive coefficient for *avgUS*, the average unit sales, means that the more the customers buy every time they come at this retailer, the more likely they are to use the coupons. This coefficient is very weak actually because of the effect size: it is coherent that a small change in the average unit sales only have a little implication on the redemption rate.

[5]For the significancy, $***$ means a p-value below 0.001, $**$ a p-value between 0.001 and 0.01 and $*$ a p-value between 0.01 and 0.1

Variable	Coeff	Std Error	Sign.
Intercept	-5.85	0.05	***
priceOff	4.70	0.12	***
saturday	0.41	0.09	***
evening	0.09	0.03	**
morning	-0.17	0.03	***
cpTop	2.63	0.06	***
ist	0.01	0.00	***
avgUS	0.016	0.00	***
priceOff × *saturday*	-0.73	0.26	**

Table 9: Output of the best model - Time dimension

- Finally, the last product of variables, *priceOff* × *saturday* and its negative coefficient indicates that customers are less sensitive to the amount of the discount on Saturdays, compared to the other days of the week. It is worth adding that even though this last coefficient is negative, it does not change the fact that the overall effect on *saturday* is positive, since the *priceOff* variable does not exceed 0.4.

The signs of the coefficients allow a qualitative interpretation. However, for a quantitative approach, we need to compute the odd-ratios.

Variable	Odd ratio	Inf	Sup
Intercept	0.0029	0.0026	0.0032
priceOff	109.86	87.30	180.60
saturday	1.51	1.26	1.81
evening	1.10	1.02	1.17
morning	0.84	0.80	0.89
cpTop	1.39	12.27	15.78
ist	1.01	1.01	1.01
avgUS	1.01	1.01	1.02
priceOff × *saturday*	0.48	0.29	0.81

Table 10: Odd-ratios and their confidence interval for the best model - Time dimension

From this table, we can deduce that:

- if the coupon is printed on an evening, the probability of redemption increases by 10% compared to if it is printed on an afternoon, all other parameters remaining the same
- if the coupon is printed on a morning, the probability of redemption decreases by 16% compared to if it is printed on the afternoon, all other parameters remaining the same
- if the average unit sales is increased by 10 euros, the probability of redemption increases by 17%, all other parameters remaining the same[6].

[6] since $(e^{0.016})^{10} \simeq 1.17$

- if the price-off is increased by 10 percentage point on a week day, the probability of redemption increases by 60%, all other parameters remaining the same[7].
- if the price-off is increased by 10 percentage point on a Saturday, the probability of redemption increases by 49%, all other parameters remaining the same[8].
- if the coupon is printed on a Saturday, then the increase in the probability of redemption depends on the price-off. For instance, considering a price-off of 20%, this probability would increase by 30%, all other parameters remaining the same[9].

5.4 Consequences for the recommender engine

The above results encourage to take more into account the time dimension into the model to better predict the redemption of the coupons.

The magnitude of the improvement of the model is certainly not huge but it still can make the predictions more accurate and better allocate the coupons to the customers. And most important, the new model takes into account new parameters that may change the optimization results.

A first direct consequence would probably be that discount on Saturdays would be on average lower than they are nowadays. Two reasons for that:

- customers are globally more likely to use the coupons on Saturdays
- customers are less sensitive to the price-off on Saturdays, than they are during the other days of the week.

Another consequence would be that a same customer could get a better discount on the same product coming on the morning instead of the afternoon or the evening. He/She could also get a higher price-off if he/she comes on a Thursday instead of a Saturday.

[7] since $\sqrt[10]{109.86} \simeq 1.60$
[8] since $\sqrt[10]{e^{4.70-0.73}} \simeq 1.49$
[9] since $e^{0.41-0.2\times 0.73} \simeq 1.30$

6 The spatial dimension

As in the previous part, we start with an exploratory analysis to perceive the first potential insights since it could give some guidance for the model improvement.

6.1 Exploratory analysis

Regarding the *bezirk* first, we clearly see that for some *bezirks* the redemption rate is higher.

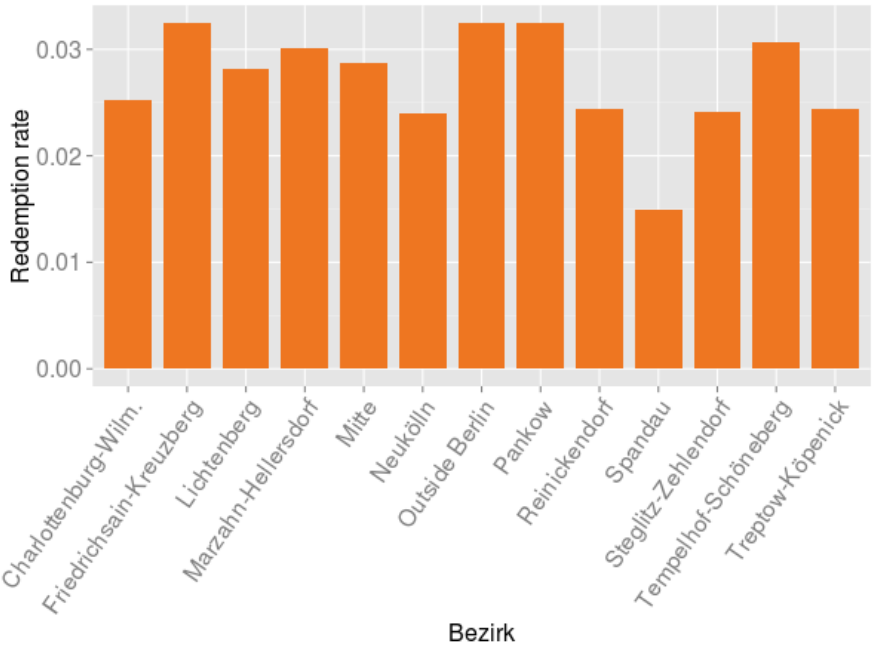

Figure 12: Redemption rate by *bezirk*

However, we clearly see no obvious correlation with the wealth of the respective *bezirk*.

The three *bezirke* with the highest net income per inhabitant in 2011, by decreasing order, are *Steglitz-Zehlendorf*, *Pankow* and *Charlottenburg-Wilmersdorf*. The three with the lowest one are, by increasing order: *Neukölln*, *Mitte* and *Kreuzberg-Friedrichshain* [10]. However, *Neukölln*, *Mitte* and *Kreuzberg-Friedrichshain* have very distinct redemption rates for instance, whereas they have similar net income per inhabitant. On the contrary, *Pankow* and *Kreuzberg-Friedrichshain* have a similar redemption rate whereas they lie on the two extreme sides of the income ranking. Another example is that *Mitte* and *Marzahn-Hellersdorf* have a similar redemption rate whereas they are both completely different. The first one is not so residential but more like a business district in some parts, very commercial and lively in other parts. On the contrary, the other one is very residential.

Regarding the geographical position of the shops relatively to the wall during the time Berlin had been split in two distinct parts, we see that in the East, the redemption rate is much higher: 3.0% vs. 2.4% in the West.

[10] Source: *Basisdaten zur Bevölkerung und sozialen Lage im Bezirk Mitte*, Bezirksamt Mitte von Berlin, January 2013

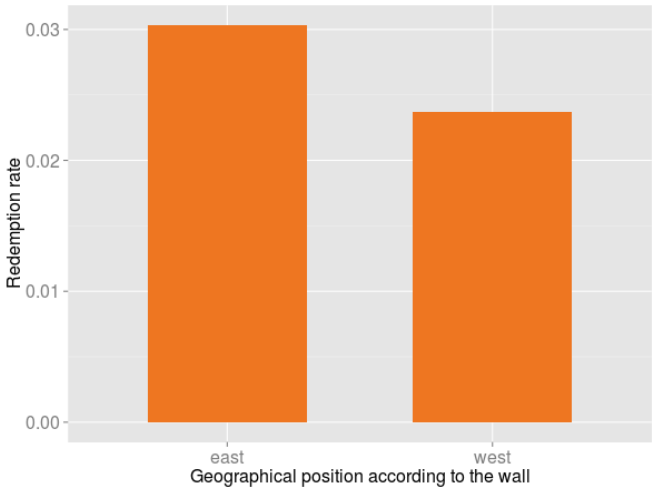

Figure 13: Redemption rate according to the Eastern/Western location

However, looking further, it is noticeable that the sample sizes differ quite a lot as shown in the graph below: more than 740000 observations for the Eastern shops versus only approximately 127000 for the Western ones. The main reason being that the 31st shops where the system was initially implemented are mainly located into the East part of Berlin.

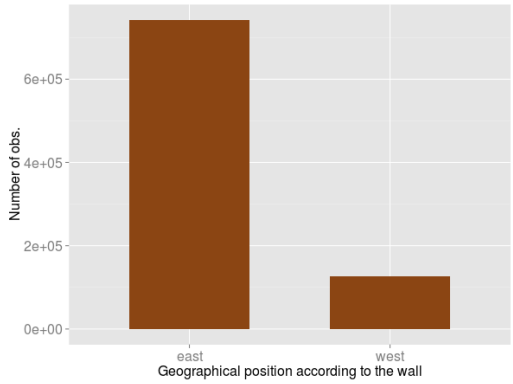

Figure 14: Number of observations according to the Eastern/Western location

Nonetheless, a Welch t-test tells us that the difference is significant and the 95% confidence interval (in percentage points) is: $[0.57; 0.76]$.

Finally, when checking the redemption rate in central areas (i.e. within the Berlin *Ring*) against areas further from the center, we cannot see any significant difference.

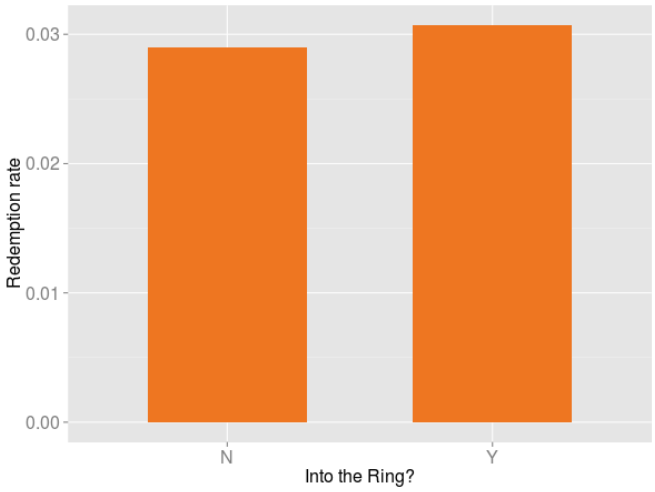

Figure 15: Redemption rate according to the location regarding the Berlin *Ring*

This is the same statement for most of the variables related to competitor. Two examples are displayed below: type of the closest competitor and number of competitors in a radius of 200 meters.

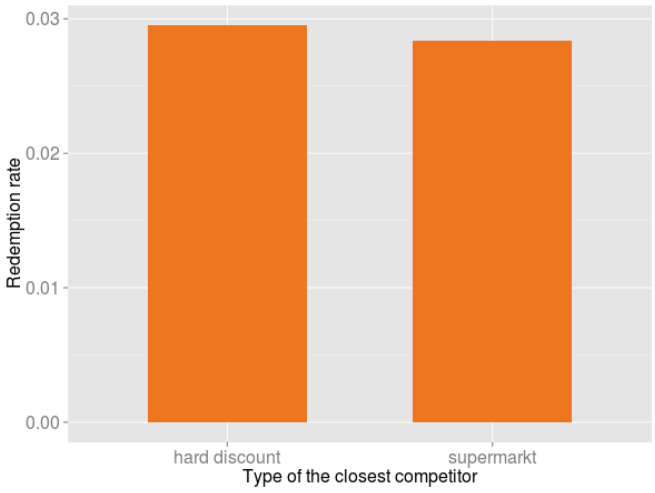

Figure 16: Redemption rate according to the type of the closest competitor

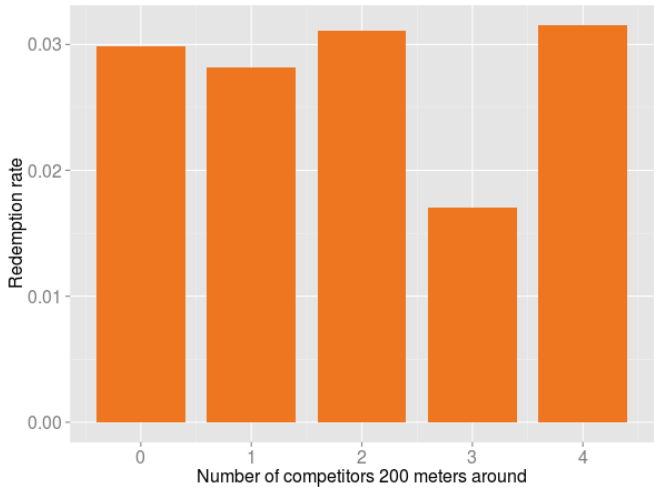

Figure 17: Redemption rate according to the number of competitors in a radius of 200 meters

However, one exception is with the number of competitors in a radius of 500 meters.

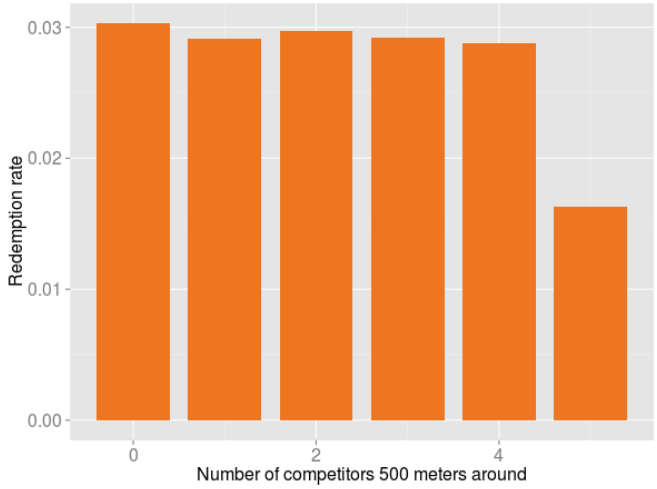

Figure 18: Redemption rate according to the number of competitors in a radius of 500 meters

It seems like the more competitors there are in such a radius, the less the redemption rate is. This could make sense since the competitive pressure is higher with more competitors and so customers who are very price-sensitive would go to cheaper places anyway. Then the customers going to *our* supermarket would not be so interested in the coupon system. Though, this is the only indicator showing a sort of pattern and it may just be an exception.

This lack of patterns and insights can actually be summarized into a graph showing the spatial repartition of the different shops where the recommender engine is operating together with their respective redemption rate. Here again, there seems to be no specific logic and no pattern.

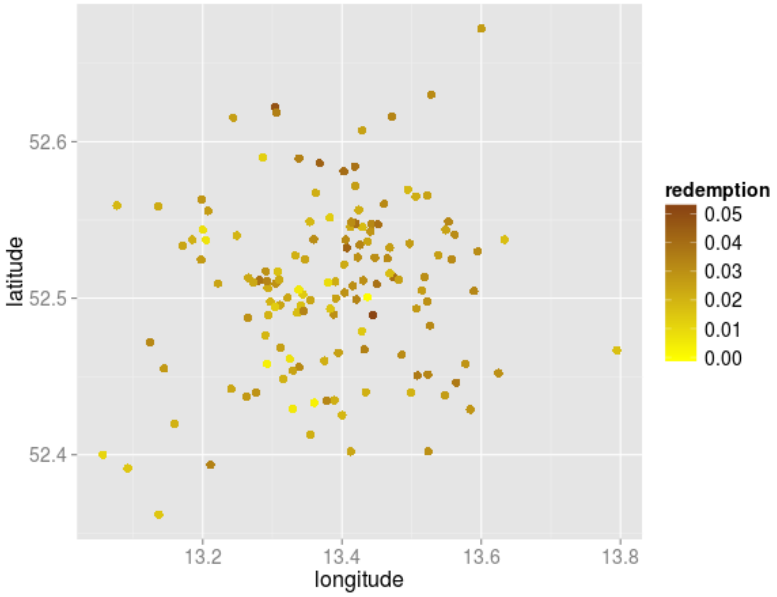

Figure 19: Spatial repartition of the shops and their respective redemption rate

6.2 Introducing the spatial dimension into the model

As the dataset changed compared to the previous section, the base model we consider does not have the same performance indicators as before:

Model	**Gini coeff**	**Brier**
$redemption \sim priceOff + cpTop$	0.364	0.029

Table 11: Base model - Spatial dimension

As pointed out above, the geographical indicators do not seem to differentiate a lot the redemption rate. This is confirmed when adding them to the base logistic model. Indeed they do not bring enough information to the model to be kept.

Below are a couple of models and their resulting Gini coefficient. No significant improvement can be observed compared to the base model.

Attempts to add some variables related to the customers' behaviour were also unsuccessful, as it has been done as well for the time dimension. The time-related variables were also used for some models to see whether time and spatial dimensions could interfere together. However, no improvement compared to the base model could be observed for every new attempt.

Model	Gini coeff	Brier
$redemption \sim priceOff + cpTop + into\_ring$	0.363	0.029
$redemption \sim priceOff + cpTop + bezirk$	0.366	0.029
$redemption \sim priceOff + cpTop + east\_west$	0.364	0.029
$redemption \sim priceOff + cpTop + nb\_competitors200$	0.363	0.029
$redemption \sim priceOff + cpTop + nb\_competitors500$	0.363	0.029
$redemption \sim priceOff + cpTop + nb\_competitors1000$	0.364	0.029
$redemption \sim priceOff + cpTop + closest\_competitor$	0.363	0.029
$redemption \sim priceOff + cpTop + min\_competitor$	0.363	0.029
$redemption \sim priceOff + cpTop + min\_hard\_discount$	0.363	0.029
$redemption \sim priceOff + cpTop + min\_supermarkt$	0.363	0.029

Table 12: Unsuccessful models - Spatial dimension

6.3 Additional analyses

In order to try to get some insights about the influence of the spatial dimension on the redemption, some further analyses were tested.

6.3.1 Considering the nature of the discounted products

First, an attempt was made to sub-sample the dataset according to the category of the discounted product. The idea behind is that maybe, some products which cannot be found in every supermarkets oblige the customers to adopt a different behaviour than products which are sold everywhere. More concretely, some brands are not available in hard-discount shops and some people would not buy some products in hard-discount retailers because of a poorer quality. So for this kind of product, there could be different spatial parameters, especially related to competitors, which may influence the redemption.

To proceed, the different campaigns were connected to categories according to the discounted product. The data was then split according to these categories. Sub-samples include either one of the following category or a meaningful combination of few of these sections. 11 different categories were chosen:

- Alcoholic beverages
- Beer
- Dairy products
- Fruits and vegetables
- Home care
- Non-alcoholic beverages
- Other food
- Pets
- Sweet products
- Tea or coffee
- Mineral water

Then for each of the subsets, several attempts were made to include the built geographical variables and indicators. This has been implemented through a function, doing similar estimations as before with the whole sample but with reduced data. This approach was actually just sub-sampling compared to what was done just before.

As a result, for every sub-sample, the base model was not significantly improved while introducing geographical variables and indicators. Furthermore, for some categories, there were too few observations to really assess the models. This was for instance the case for *Tea or coffee* and *Mineral water*.

So taking into consideration the discounted products do not allow either to conclude to an influence of spatial parameters on the redemption. It seems like whatever the different discounted products were, the spatial parameters did not interfere with the redemption.

6.3.2 More advanced models

Then, some more complex models were implemented to attempt to get further insights on the customer behaviour.

Neural networks

The neural networks may be a good method to get more information about the patterns in location-related data and their implications for the redemption.

Neural networks tend to replicate the functioning of the human brain.

- They consider the different independent variables as inputs.
- The different inputs are aggregated to form *hidden layers*.
- From these *hidden layers*, a non-linear function transforms the aggregated data into probabilities.

Several sets of variables were tried to build such a model:

- First, a program was run to get the optimal input parameters for the model: the number of hidden nodes and the weight decay[11]. This loop then gives back the parameters with which the AUC is maximized.
- Afterwards, the optimal parameters were set up to estimate the weights, the AUC and the representation to observe relationships between the different variables and the redemption.

However, partly due to a lack of time, any better model was reached. Everytime, the Gini was equivalent to or below 0.36 - the AUC for the base model.

An example of a neural network representation is shown below. The input layer (nodes I) are, from top to bottom: *priceOff*, *cpTop*, *hour*, *into_ring*, *closest_competitor*, *nb_competitors200* and *east_west*. There are three hidden layers (nodes H) and two biased layers (nodes B).

The conclusion of this approach is still that the price-off (*node I1*) as well as the customers' characteristics (*node I2*) get super high weights compared to the spatial parameters.

[11]The weight decay is a parameter to constrain the weight not to be too high, which could allow very complex models

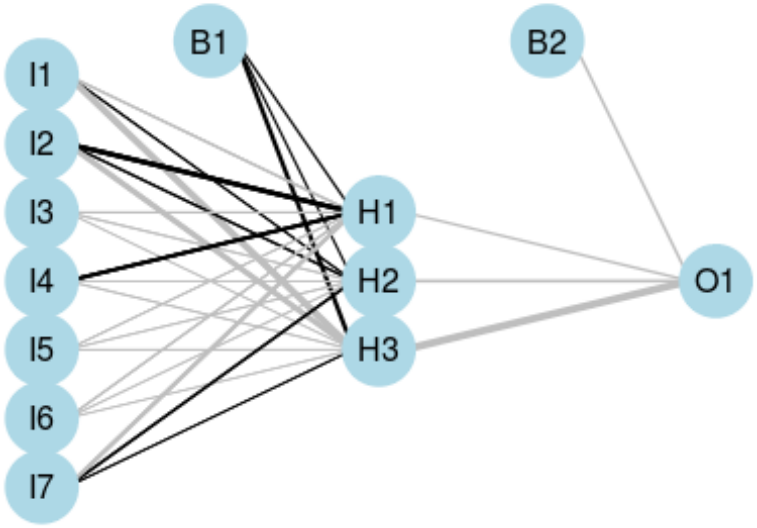

Figure 20: Graphical representation of a neural network modelling redemption

How to read this graph: the connections between the nodes (neurons) have their width proportional to the weight between the nodes and the color indicates the sign (black for positive and grey for negative).

Classification models

Partitioning classification was also implemented. Briefly, this kind of model works as follows:

- it starts by considering the sample as one class, called the *root node*.
- it looks for the variable or the set of variable that allows to split the all sample in a way that it improves the predictive power. More concretely, it looks for how to split the data is several groups, called *internal nodes*, in which the purity is better than in the whole sample. The purity is a concept that describes the homogeneity of a sample of observation with regards to the target variable: a node is said to be *pure* if it contains only observations from the same class. In our case, it would be the *redemption* classes.
- This process is repeated until the purity cannot be improved anymore.

However, the problem is that the redemption classes are very unbalanced. Indeed, more than 97% of the coupons were unused (no redemption). Then, this kind of model does not work since it is very unlikely that the classification can be improved in comparison with considering that none of the coupons was used - so getting more than 97% of accuracy. The *root node* is actually already fairly pure.

6.4 Consequences for the recommender engine

The above results lead us not to take into account the spatial dimension into the model. Indeed, this would not allow to better predict the redemption of the coupons with the logistic regression, but only make the model more complex.

More precisely, it tells us that central-district inhabitants do not have such a different redemption behaviour compared to their fellows outside the *Ring*. Same assessment for people going to shops in the Eastern part versus those going in the Western part.

Then it seems that in every part of Berlin, the redemption is predictable in a similar way: taking into account the price-off, knowing the customer's behaviour and finally considering the time of the coupon printing. This is also always the case while taking into account several different competitive environments. Whatever the nearest competitor is, very close or not, the customer redemption probability will not be affected.

To a broader extent, it can be induced that finally, customers do not care that much about the shops around the retailer where they usually go. They are simply not permanently optimizing their cart according to the shops around them every single time they need to go shopping.

7 Conclusions

As a reminder, the initial purpose of this project was to improve the predictive performance for the attribution model of the recommender engine. Better predictions would truly allow to optimize the recommender engine while keeping the budgetary constraints in mind.

The time dimension was successfully introduced in the model and in this sense, the objective was reached. The model performance was indeed improved through the introduction of some time-related variables. These variables not only improve the model but also teach us about the customers' behaviour: redemption is thus more likely to happen on Saturdays than during weekdays and customers tend to use more their coupons on the evenings. These learnings should obviously be taken into account in order to improve the recommender engine.

On the other hand, the introduction of the spatial dimension did not lead to any significant improvement of the predictive performance of the model. Still, this part of the project was useful to learn that finally, redemption behaviour does not significantly vary according to geographical parameters. This means that the competitive environment as well as the surroundings' characteristics do not have any significant impact on the probability for customers to use their coupons. There is therefore no point in taking spatial parameters into the attribution model, since it would only increase its complexity.

The results of this project give a good overview of the interference between time and spatial parameters and redemption. However, there is still room for improvement.

The time analysis could not take into consideration larger time dimensions than days, due to the short-period during which the data were collected. It was then not possible to consider differences in redemption behaviour across weeks and months. Yet, customers do change their shopping habits according to the period of the year. Regarding redemption behaviour, it is for instance legitimate to wonder: do customers use more the coupons in high-expenses times (e.g. Christmas period)? On a monthly dimension, do they use them more at the end of the months when they have to be more careful with the expenses? These questions are worth being considered.

On the geographical side, the data were at the level of one city - even if some stores are right outside of Berlin. Although this city is rather big with some geographical divergences, it does not provide a very rich collection of geographical characteristics since the different stores are not located in very disparate areas. That is why it could be worth extending the analysis on a larger geographical area, in case the recommender engine is set up in other cities and regions.

Appendix

Detailed list of variables in the initial database

Variable	Type	Meaning
Redemption information		
userId	character	Unique loyalty card user index
campaignId	character	Unique index for each promotion campaign
redemption	integer	Binary indicator of coupon redemption
priceOff	numeric	Price off in percentage
Time information		
printDate	character	Date of coupon print
printTime	integer	Time of coupon print (in seconds since midnight)
printWeekDay	character	Day of week of coupon print
Store information		
storeId	character	Unique store index
storeIdImputed	integer	Is the store information imputed
storeName	character	The name (of the street) of the store
zipCode	integer	The zip code where the store is located
User information (derived from purchase history)		
ipt.N	integer	Number of coupon prints
ipt	numeric	Inter print time i.e. avg. number of day between coupon prints
irt.N	integer	Number of coupon redemption
irt	numeric	Inter redemption time i.e. avg. number of day between coupon redemptions
ist.N	integer	Number of card scans at the till
ist	numeric	Inter scan time i.e. avg. number of day between card scans
nTrips	integer	Number of purchase trips observed
totalUS	integer	Total unit sales
totalVS	integer	Total value sales i.e. unit sales * unit price
avgUS	numeric	Average unit sales
avgVS	numeric	Average value sales
cpTop	numeric	Conditional probability[a]
alcohol_ other	integer	Does user buy alcohol_ other products?
baby	integer	Does user buy ‘baby‘ products?
beer	integer	Does user buy ‘beer‘ products?
cat	integer	Does user buy ‘cat‘ products?
child	integer	Does user buy ‘child‘ products?
convenience	integer	Does user buy ‘convenience‘ products?
dairy	integer	Does user buy ‘dairy‘ products?
dog	integer	Does user buy ‘dog‘ products?
egg	integer	Does user buy ‘egg‘ products?
liquor	integer	Does user buy ‘liquor‘ products?
meat	integer	Does user buy ‘meat‘ products?
seafood	integer	Does user buy ‘seafood‘ products?
wine	integer	Does user buy ‘wine‘ products?
woman	integer	Does user buy ‘woman‘ products?

[a]for each user and campaign combination; this variables measures the base preference; higher value means higher preference

Detailed list of variables in the second database

Variable	Type	Meaning
Redemption information		
userId	character	Unique loyalty card user index
campaignId	character	Unique index for each promotion campaign
redemption	integer	Binary indicator of coupon redemption
priceOff	numeric	Price off in percentage
Time information		
printDate	character	Date of coupon print
printTime	integer	Time of coupon print (in seconds since midnight)
Store information		
storeId	character	Unique store index
storeName	character	The name (of the street) of the store
User information (derived from purchase history)		
ipt	numeric	Inter print time i.e. avg. number of day between coupon prints
avgUS	numeric	Average unit sales
cpTop	numeric	Conditional probability[a]

[a]for each user and campaign combination; this variables measures the base preference; higher value means higher preference

References

Baesens, B. (2014): *Analytics in a Big Data World*, Wiley.

Flach, P. (2004): "The many faces of ROC analysis in machine learning," *University of Bristol*.

Green, H. and G. Milne (2010): "Assessing model performance: The Gini statistic and its standard error," *Journal of Database Marketing and Customer Strategy Management*, 17, 36–48.

Hastie, T., R. Tibshirani, and J. Friedman (2009): *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer.

Lee, W. (1999): "Probabilistic analysis of global performances of diagnostic tests: interpreting the Lorenz curve-based summary measures," *Statistics in Medicine*, 455–471.

Lobo, J., A. Jimnez-Valverde, and R. Real (1988): "AUC: a misleading measure of the performance of predictive distribution models," 2789–2795.

Mason, S. and N. Graham (2002): "Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical signi cance and interpretation," *Q.J.R. Meteorol. Soc.*, 128, 2145–2166.

Power, D. (2011): "Evaluation: fron precision, recall and F-measure to ROC, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, 2, 37–65.

Declaration of Authorship

I hereby confirm that I have authored this Master's thesis independently and without use of others than the indicated sources. All passages which are literally or in general matter taken out of publications or other sources are marked as such.

Berlin, October 30, 2015

Pierre Navarro